

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 09-231347

(43)Date of publication of application : 05.09.1997

(51)Int.Cl.

G06T 1/20

H04N 1/21

H04N 1/40

(21)Application number : 08-060049

(71)Applicant : NEC CORP

(22)Date of filing : 22.02.1996

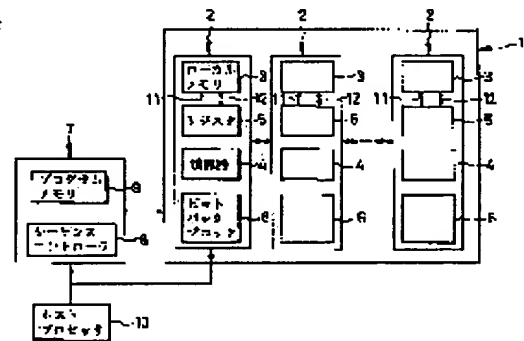
(72)Inventor : YAMASHITA NOBUYUKI

## (54) IMAGE PROCESSOR

### (57)Abstract:

**PROBLEM TO BE SOLVED:** To accelerate binary pixel processing by allowing in each processor to bit-pack the data of the prescribed register of a proximity processor and data in a prescribed register within each processor.

**SOLUTION:** Each processor 2 is provided with m-sets of 1-bit registers 5 and bit pack blocks 6 and these registers 5 can be referred to from an adjacent processor 2. Then each processor 2 assigns a (m2-1) bit obtained by adding data of m-bit from each adjacent prescribed processor 2 and the (m-1) of its own register 5 of a bit position corresponding to the stored register 5 to store in each bit register by bit-packing. In addition data is simultaneously shifted between m-sets of 1-bit registers 5 and to a processor 2 of an end part to which data is not sent at the time of shifting, next data is loaded from a local memory 3. Thereby fast image processing is attained with respect to binary image data.



## LEGAL STATUS

[Date of request for examination] 22.02.1996

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number] 2812292

[Date of registration] 07.08.1998

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平9-231347

(43)公開日 平成9年(1997)9月5日

(51)Int.Cl. <sup>8</sup>	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 T	1/20		G 0 6 F 15/66	K
H 0 4 N	1/21		H 0 4 N 1/21	
	1/40		1/40	Z

審査請求 有 請求項の数4 F D (全 10 頁)

(21)出願番号 特願平8-60049

(22)出願日 平成8年(1996)2月22日

(71)出願人 000004237

日本電気株式会社

東京都港区芝五丁目7番1号

(72)発明者 山下 信行

東京都港区芝五丁目7番1号 日本電気株式会社内

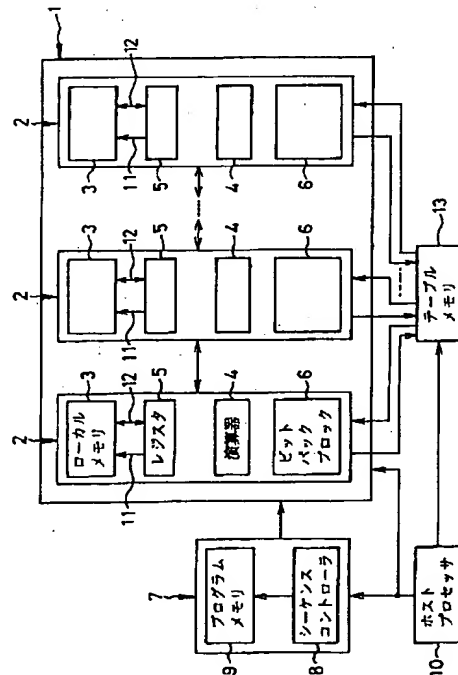
(74)代理人 弁理士 加藤 朝道

(54)【発明の名称】 画像処理装置

(57)【要約】

【課題】プロセッサを複数内蔵する並列画像処理装置において、二値画像処理を高速に行うことを可能にする。

【解決手段】ローカルメモリ3、演算器4、レジスタファイル5、ビットパック処理を行うビットパックブロック6からなるプロセッサ2を一次元状に接続したプロセッサアレイ1、テーブル参照処理を行うためのテーブルメモリ13、命令供給回路7、ホストプロセッサ10から構成される並列画像処理装置であって、プロセッサアレイ内の各プロセッサは近傍プロセッサのレジスタのデータを参照でき、その参照データと各プロセッサ内のデータをビットパックする手段を備え、ビットパックしたデータに論理演算を行うか、又はビットパックしたデータをアドレスとしてテーブルメモリ13をリードして演算結果を得る。



**【特許請求の範囲】**

【請求項1】複数のプロセッサがアレイ状に設けられる画像処理装置において、

各プロセッサは、近接プロセッサの所定のレジスタにあるデータを直接参照可能とされ、前記近接プロセッサの所定のレジスタのデータと前記各プロセッサ内の所定のレジスタにあるデータとをビットパックする手段を有することを特徴とする並列画像処理装置。

【請求項2】前記プロセッサが、前記ビットパック操作において、ビットパックに用いられるデータを格納するレジスタ間で、データをシフトすると共に、次のビットパック処理で用いられるデータをメモリからレジスタにリードする手段を備えたことを特徴とする請求項1記載の並列画像処理装置。

【請求項3】テーブル参照用メモリとして、1つのリードポートと、プロセッサ総数と同数又は前記プロセッサ総数を所定の正数で除してなる個数のリードポートとを有するメモリを備えたことを特徴とする請求項1又は2記載の並列画像処理装置。

【請求項4】ローカルメモリ、演算器、レジスタ、及びビットパック処理を行うビットパック手段を含んでなるプロセッサをアレイ状に複数接続してなる並列画像処理装置であって、

前記各プロセッサは、近傍プロセッサの画素情報を保持する所定のレジスタのデータを直接参照可能とされると共に、前記近傍プロセッサの所定のレジスタと自プロセッサ内のビットパックに用いるレジスタとを転送元（ソース）として転送先（デスティネーション）に一度にビットパックするように構成され、

ビットパックに用いられるデータを格納するレジスタ間でデータをシフトすると共に、次のビットパック処理で用いられるデータを前記ローカルメモリから前記レジスタにロードする手段と、を備えたことを特徴とする並列画像処理装置。

**【発明の詳細な説明】****【0001】**

【発明の属する技術分野】本発明は、画像処理装置に関し、特に高速に二値画像処理を行う並列画像処理装置に関する。

**【0002】**

【従来の技術】図8に、従来の並列型画像処理装置の構成を示す。図8を参照して、従来の並列型画像処理装置において、プロセッサアレイ100は、画像データおよび処理結果を蓄えるローカルメモリ3、演算器4、及びレジスタファイル5からなるプロセッサ200を一次元状に複数接続してなるものであり、命令供給回路7は、プログラムメモリ8と、シーケンスコントローラ9と、からなる。

【0003】プロセッサ200において、レジスタファイル5とローカルメモリ3とは、アドレス線11とデー

タ線12とで相互に接続されており、各プロセッサ200は、レジスタファイル5の複数個あるレジスタから、命令によって選択されたレジスタの内容をアドレスとして、ローカルメモリ3をアクセスするという間接アドレッシング機能を具備している。

【0004】各プロセッサ200は、レジスタファイル5内のあるレジスタの内容をアドレスとして、各プロセッサ内のローカルメモリ3にアクセスし、アクセスされたデータを処理し、再びローカルメモリ3に書き込む。この一連のメモリアクセスおよび処理は、全プロセッサで同時に実行することができる。

【0005】画像処理においては、常に、データが多値で処理されるというわけではなく、処理の途中において、二値化されたデータを処理する場合が頻繁にある。例えば、膨張、収縮、孤立点除去や細線化といった処理においては、二値データが処理対象となり、このような処理において、処理対象の画素の処理後の値は、処理対象画素の、例えば3×3近傍画素等の近傍画素の値を演算して決定される。

【0006】図8に示した従来の並列型画像処理装置において、ある画素の値をその近傍領域の画素値によって決定するという二値画像処理を実行する場合、次のような処理を行う。

【0007】まず、一のプロセッサは、メモリ（ローカルメモリ）から、処理に必要な画素データをロードし、次に、近傍のプロセッサから近傍のデータを転送することによって、処理対象画素を保持するプロセッサ上に近傍画素のデータを集め、集められたそれぞれの1ビットデータに対して各種論理演算を行うことによって、処理対象画素の処理後の値を決定し、最後にそのデータ（処理後の値）をメモリに書き込む。

【0008】この処理において、集められた1ビットデータを、一旦、プロセッサ毎にビットパックして、多ビット化したのち、ビットパックされたデータに対して論理演算処理を施すことにより、1ビット毎に論理演算を行う場合に比べて、高速に処理が行えることにもなる。

【0009】また、ビットパックデータに対して論理演算処理を実行する代わりに、ビットパックデータをアドレスとしてメモリ（すなわちテーブル）をアクセスし、当該アドレスに予め格納されているデータ（論理演算結果）を読み出すというテーブル参照処理を行うことによって、高速に処理できる場合もある。このテーブル参照処理を行う際には、各プロセッサ内のローカルメモリをテーブルデータ格納用のメモリに割り当て、割り当てられたメモリに対して、上記した間接アドレッシングでメモリアクセスを行う。

**【0010】**

【発明が解決しようとする課題】しかしながら、上記した従来の並列型画像処理装置においては、二値画像データをビットパックする際に要する処理時間が長くなり、

高速処理の実現を阻止しているという問題点を有している。これは、近傍画素値をプロセッサ間転送によって、1ビットずつ転送していることと、集められた画素データに対しても、シフト演算等によって、所望のビット位置にまでデータを持って来た後に、論理演算を行い、ビットパックするという具合に、ビットパックするのに、多くの演算を必要とすることによる。

【0011】また、上記した従来の並列型画像処理装置においては、各プロセッサで同じテーブルデータを用いる場合には、同じテーブルデータを各プロセッサ毎に持つことになり、ローカルメモリを無駄に消費してしまう、という問題点を有している。

【0012】従って、本発明は、上記問題点を解消すべく為されたものであって、その目的は、二値画素処理を、従来の装置よりも高速に行うことができる並列画像処理装置を提供することにある。

#### 【0013】

【課題を解決するための手段】前記目的を達成するために、本発明は、複数のプロセッサがアレイ状に設けられてなる画像処理装置において、各プロセッサは、近接プロセッサの所定のレジスタにあるデータを直接参照可能とされ、前記近接プロセッサの所定のレジスタのデータと前記各プロセッサ内の所定のレジスタにあるデータとをビットパックする手段を有することを特徴とする並列画像処理装置を提供する。

【0014】本発明は、前記プロセッサが、前記ビットパック操作において、ビットパックに用いられるデータを格納するレジスタ間で、データをシフトすると共に、次のビットパック処理で用いられるデータをメモリからレジスタにリードする手段を備えたことを特徴とする。

【0015】本発明においては、テーブル参照用メモリとして、1つのライトポートと、プロセッサ総数と同数又は前記プロセッサ総数を所定の正数で除してなる個数のリードポートとを有するメモリを備えたことを特徴とする。

【0016】本発明の原理・作用を以下に説明する。本発明は、複数のプロセッサから構成されたプロセッサアレイが、画像メモリに対して1度に、画像の行単位のデータのリード/ライトを実行することができる、プログラム実行可能な並列画像処理装置において、二値画像データをブロック単位でビットパックする手段を備え、さらに、ビットパック処理と同時に、すなわち、好ましくはビットパック処理と同じ命令サイクル内に、次にビットパックするデータをメモリからロードする手段を具備したことにより、ビットパック演算を高速化すると共に、回路規模の増大を抑止し、二値画像データに対して高速な画像処理を実行できるようにしたものである。さらに、ビットパックされたデータによるテーブル参照処理に使用されるメモリとして1ライト/マルチリードポート構成のメモリを用い、全プロセッサからリードでき

るようにすることにより、各プロセッサのローカルメモリをテーブルメモリとして用いる場合に比べて、大幅にメモリ使用量を削減できる。

#### 【0017】

【発明の実施の形態】本発明の実施の形態を図面を参照して以下に詳細に説明する。図1は本発明の画像処理装置の一実施形態の構成を示す図である。

【0018】図1を参照して、プロセッサアレイ1は、ローカルメモリ3、演算器4、レジスタファイル5、及びビットパック処理を行うビットパックブロック6からなるプロセッサ2を一次元状に複数接続したものである。また、図1において、命令供給回路7は、シーケンスコントローラ8とプログラムメモリ9とから構成され、シーケンスコントローラ8が生成したプログラムアドレスによって、プログラムメモリ9から命令を読み出し、その読み出された内容をプロセッサアレイ1に命令として供給する。ホストプロセッサ10は、プログラムメモリ9やローカルメモリ3へのデータロード等を行う。

【0019】本実施形態の画像処理装置は、プログラム実行可能な装置であり、同一の命令ストリームで全てのプロセッサ2（プロセッサエレメント、「PE」ともいう）を動作させるSIMD（single instruction multiple data stream；単一命令複数データ制御）型並列画像処理装置とされる。この画像処理装置は、算術論理演算命令、シフト命令、フラグ操作命令、PE間データ転送命令と、ロード/ストア命令等の命令セットを有している。

【0020】図2（A）は、画像の第j行、第i列の画素を $P(i, j)$ としたときの、 $P(i, j)$ を中心画素とした $3 \times 3$ 近傍領域を示す図である。いま、各プロセッサ2のローカルメモリ3には、画像の1列のデータが1画素ずつ分散されて格納されているものとする。すなわち、i番目のプロセッサ2は、画像の第i列のデータをローカルメモリ4内に格納しながら処理を進めて行く。以下ではプロセッサ2をPE（プロセッサエレメント）、ローカルメモリをM、プロセッサの総数をn、プロセッサエレメントPEの番号をiで表し、i番目のPEを $PE(i)$ 、i番目のPEのローカルメモリの第j行を $M(i, j)$ にて表すものとする。

【0021】画像の第j行のデータは、ローカルメモリの第j行に格納されているとすると、ローカルメモリ $M(i, j)$ は、画素 $P(i, j)$ を保持することになる。

【0022】いま、i番目のプロセッサエレメントPE(i)で処理する画素を $P(i, j)$ とすると、 $P(i, j)$ を中心とする近傍 $3 \times 3$ 画素は、 $P(i-1, j-1)$ 、 $P(i, j-1)$ 、 $P(i+1, j-1)$ 、 $P(i-1, j)$ 、 $P(i, j)$ 、 $P(i+1, j)$ 、 $P(i-1, j+1)$ 、 $P(i, j+1)$ 、 $P(i+1, j+1)$ 、

( $i+1, j+1$ )である。なお、図2では、中心画素  $P(i, j)$  を  $P(k, m)$ 、すなわち  $i=k, j=m$  として示している。

【0023】これらの9画素から処理対象画素（中心画素）である  $P(i, j)$  を除く8画素を、図2(B)に示すように、8ビットデータにビットパックする場合について、以下説明を行う。

【0024】図3は、 $3 \times 3$ 領域のビットパック処理を行う構成を示した図であり、ビットパック処理を行うプロセッサエレメント  $PE(i)$  のレジスタとその両隣りの  $PE(i-1)$ 、及び  $PE(i+1)$  の接続関係が示されている。なお、図3では、番号  $i$  を  $k$  として示している。

【0025】 $PE(i)$  において、 $3 \times 3$ 領域のビットパックをする場合、左側のプロセッサ  $PE(i-1)$  から  $P(i-1, j-1)$ 、 $P(i-1, j)$ 、 $P(i-1, j+1)$  の3ビットのデータを、右側のプロセッサ  $PE(i+1)$  から  $P(i+1, j-1)$ 、 $P(i+1, j)$ 、 $P(i+1, j+1)$  の3ビットのデータを、 $PE(i)$  は必要とする。これと同様に、左側の  $PE(i-1)$  は、 $PE(i-2)$  と  $PE(i)$  からそれぞれ3ビットずつのデータを、右側の  $PE(i+1)$  は、 $PE(i)$  と  $PE(i+2)$  からそれぞれ3ビットずつのデータを必要とする。

【0026】すなわち、 $PE(i)$  については、その左右のプロセッサ  $PE(i-1)$ 、 $PE(i+1)$  とから、3ビットずつのデータ入力線と3ビットずつのデータ出力線が設けられる。

【0027】かかる構成は、全  $PE$  について共通であるため、ビットパックのために、各  $PE$  は、3ビットのデータ入力線と3ビットのデータ出力線の6ビットのデータ転送線20が、隣接  $PE$  との接続において設けられるものとする。

【0028】そして、各  $PE$  は、 $r1, r2, r3$  の3つの1ビットレジスタ21を持ち、これら3つのレジスタは、隣接  $PE$  から直接参照可能に構成されている。

【0029】このため、データ転送命令によって隣接した  $PE$  間でデータ転送を行うことなく、各  $PE$  は、両隣の  $PE$  の3つの1ビットレジスタ  $r1, r2, r3$  を、ソース（転送元）として用いることができる。すなわち、ビットパック処理時に、自  $PE$  の転送先（デスティネーション）のレジスタ又はメモリ等へ、両隣の  $PE$  の3つの1ビットレジスタ  $r1, r2, r3$  をソースオペランドとしても指定することができる。

【0030】さらに、図3に示すように、1ビットレジスタ  $r1 \sim r3$  は、各  $PE$  内において、セクタ22によって、1ビット論理左シフトと、1ビット論理右シフトと、が行えるように構成されている。ここで、 $PE(i)$  の1ビットレジスタ  $r1, r2, r3$  をそれぞれ、 $r1(i), r2(i), r3(i)$  とする。

【0031】まず、メモリの  $(j-1)$  行目のデータに対するロード命令によって、メモリからプロセッサアレイへ  $(j-1)$  行目のデータが、各  $PE$  の  $r1$  に転送される。

【0032】次に、同様にして、ロード命令によって、メモリの  $j$  行目、 $(j+1)$  行目のデータが、各  $PE$  の  $r2, r3$  に転送される。

【0033】この時点で、 $PE(i)$  の1ビットレジスタ  $r1(i)$  は、 $M(i, j-1)$  を、 $r2(i)$  は、 $M(i, j)$  を、 $r3(i)$  は、 $M(i, j+1)$  を保持する。

【0034】ここで、 $PE(i)$  のデスティネーション・レジスタ23を  $rd(i)$  としてビットパック命令を行うと、 $rd(i)$  のMSB（ビット7）からLSB（ビット0）の順に、 $r1(i-1), r1(i), r1(i+1), r2(i-1), r2(i+1), r3(i-1), r3(i), r3(i+1)$  の内容が格納される。

【0035】これによって、中心画素  $P(i, j)$  を中心とした  $3 \times 3$  画素領域の2値データの内、中心画素である  $P(i, j)$  を除く8画素のデータが  $rd(i)$  にビットパックされて格納される。

【0036】上記のビットパック処理は、 $\{PE(i) : 0 \leq i \leq n-1\}$  の  $n$  個の  $PE$  のうち、 $1 \leq i \leq n-2$  の範囲にある  $PE(i)$  で同時に実行される。

【0037】一次元プロセッサアレイ1の両端の  $PE$  については、例外処理が行われるが、図4は、この両端の  $PE$  でのビットパック処理の動作の説明をする図である。

【0038】 $n$  個のプロセッサエレメントの一次元アレイからなるプロセッサアレイの両端の  $PE(0)$  と  $PE(n-1)$  について、それぞれ、 $PE(-1)$ 、 $PE(n)$  が存在しないので、用途に応じて、①  $PE(0)$  と  $PE(n-1)$  を隣接  $PE$  とみなしてデータを必要とする場合、②隣接  $PE$  が存在しない  $PE$  方向からはデータとして「0」を入力する必要がある場合、③隣接  $PE$  が存在しない  $PE$  方向からはデータとして「1」を入力する必要がある場合、の3通りの場合が代表的である。

【0039】そこで、本実施形態においては、予めモードレジスタ25に、命令によって、前述した3通りのモードのうちの1つのモードに設定しておき、その設定された2ビットの値を選択信号とするセクタ24によって、 $PE(0)$  については、 $PE(n-1)$  の1ビットレジスタ  $r1, r2, r3$  の値か、「0」又は「1」を選択可能とし、 $PE(n-1)$  についても同様に、モードレジスタ25からの出力を選択信号とするセクタ24によって、 $PE(0)$  の1ビットレジスタ  $r1, r2, r3$  の値か、「0」又は「1」を選択可能としている。

【0040】第  $j$  行の各画素を中心とする  $3 \times 3$  画素領

域のビットパック処理が終了して、次の第 $(j+1)$ 行について処理を行う際には、その前後の第 $j$ 行と第 $(j+2)$ 行のデータが必要になる。

【0041】第 $j$ 行と第 $(j+1)$ 行のデータは、既にそれぞれ1ビットレジスタ $r_2$ と $r_3$ に保持されており、このデータを利用することができ、また1ビットレジスタ $r_1$ に格納されている第 $(j-1)$ 行のデータは不要となる。ここで、ロード命令によって、第 $(j+2)$ 行のデータを1ビットレジスタ $r_1$ に転送すると、最初のビットパック処理では、中心画素が $r_2$ に、(中心画素の行数-1)行が $r_1$ に、(中心画素の行数+1)行が $r_3$ に保持されていたのが、次のビットパック処理では、中心画素が $r_3$ に、(中心画素の行数-1)行が $r_2$ に、(中心画素の行数+1)行が $r_1$ に保持される。

【0042】この $r_1$ 、 $r_2$ 、 $r_3$ を用いて、ビットパック処理を行うためには、デスティネーション・レジスタ $rd(i)$ のMSB(最上位ビット)からLSB(最下位ビット)の順に、 $r_2(i-1)$ 、 $r_2(i)$ 、 $r_2(i+1)$ 、 $r_3(i-1)$ 、 $r_3(i+1)$ 、 $r_1(i-1)$ 、 $r_1(i)$ 、 $r_1(i+1)$ の内容を格納する必要があり、 $rd(i)$ の各ビット位置には、最初のビットパック処理とは異なるレジスタが格納されることになる。

【0043】さらに、その次の第 $(j+2)$ 行を中心とする $3 \times 3$ 画素領域のビットパック処理についても、第 $j$ 行、第 $(j+1)$ 行を中心とする $3 \times 3$ 画素領域のビットパック処理の場合とは異なる処理が必要になる。

【0044】この3通りのビットパック処理を実現するための回路としては、ビットパックデータの各ビット位置の結果を求めるために、1ビットレジスタ $r_1$ 、 $r_2$ 、 $r_3$ から1つを選択する3入力1出力セレクト回路を、ビットパックされる1語(8ビット)に対応して8個を必要とする。

【0045】ここでは、 $3 \times 3$ 画素領域の場合について説明したが、一般に、 $m \times m$ 画素領域の中心画素を除いた画素のビットパック処理を、上記の処理方法で行う場合、 $m$ 種類のビットパック命令と、 $(m^2-1)$ 個の $m$ 入力1出力セレクト回路が必要となり(これを「手法A」という)、必要な命令の数および回路規模的にも大きくなる。

【0046】本実施形態においては、この問題を解消するために、ビットパック処理と同じ命令サイクル内で、各PEの1ビットレジスタ $r_1$ 、 $r_2$ 、 $r_3$ 間で、内容の転送を行い、さらに次の処理に必要なデータをメモリから読み出すものとする。より具体的には、ビットパック命令は、3種類用意されており、1つは、上記のビットパック動作と同時に、 $r_1 \leftarrow r_2$ 、 $r_2 \leftarrow r_3$ とコピーし、さらにメモリのデータを $r_3$ に転送する『バック&シフトアップ&ロード命令』であり、もう1つは、上

記のビットパック動作と同時に、 $r_1 \rightarrow r_2$ 、 $r_2 \rightarrow r_3$ とコピーし、さらにメモリのデータを $r_1$ に転送する『バック&シフトダウン&ロード命令』であり、残りの1つは、上記のビットパック動作のみを行う『バック命令』である。これらは、処理目的に応じて使い分けられる。

【0047】図5(A)は、処理を画像の行方向に増加させながら、進めていく場合に、『バック&シフトアップ&ロード命令』実行後に、レジスタに格納される行数を示した図である。

【0048】『バック&シフトアップ&ロード命令』では、処理を画像の行方向に増加させながら進めていく場合、ビットパックすると同時に、次の行のデータが、1ビットレジスタ $r_3$ へロードされ、レジスタ $r_1$ は常に(処理対象画素の行-1)行のデータを、レジスタ $r_2$ は処理対象画素の行のデータを、レジスタ $r_3$ は(処理対象画素の行+1)行のデータを格納する。このため、次の『バック&シフトアップ&ロード命令』を即座に実行することができる。

【0049】図5(B)は、処理を画像の行方向に減少させながら、進めていく場合に、『バック&シフトダウン&ロード命令』実行後に、レジスタに格納される行数を示した図である。

【0050】『バック&シフトダウン&ロード命令』では、処理を画像の行方向に減少させながら進めていく場合、ビットパックすると同時に次の行のデータがレジスタ $r_1$ へロードされ、レジスタ $r_1$ は常に(処理対象画素の行-1)行のデータを、 $r_2$ は処理対象画素の行のデータを、 $r_3$ は(処理対象画素の行+1)行のデータを格納する。このため、次の『バック&シフトダウン&ロード命令』を即座に実行できる。さらに、レジスタ $r_1$ 、 $r_2$ 、 $r_3$ に格納されるデータの行と処理対象画素の行との関係は、『バック&シフトアップ&ロード命令』の場合と同じであるため、テーブル参照処理をする際のテーブルデータとしては、同一のものを使用できる。

【0051】本実施形態では、 $3 \times 3$ 画素領域のビットパック処理を行う場合、命令の種類は3種類と、前記手法Aと同じであるが、回路的には、3つの3入力1出力セレクトだけで済み、回路規模を削減している。

【0052】また、 $m \times m$ 画素領域のビットパック処理を行う回路では、ビットパック処理と同時に、『バック&シフトダウン&ロード命令』では、 $r_1 \leftarrow r_2$ 、 $r_2 \leftarrow r_3$ 、…、 $r_{(m-1)} \leftarrow r_m$ とコピーし、さらにメモリのデータを $r_m$ へロードすればよく、『バック&シフトアップ&ロード命令』では、 $r_1 \rightarrow r_2$ 、 $r_2 \rightarrow r_3$ 、…、 $r_{(m-1)} \rightarrow r_m$ とコピーし、さらにメモリのデータを $r_1$ へロードすればよい。

【0053】したがって、ビットパック命令は3つでよく、これは $3 \times 3$ 画素領域のビットパック処理の場合と

同じである。回路については、 $m \times m$ 画素領域のビットバック処理に対する前記手法Aの場合には、 $m$ 入力1出力セクタ回路を $(m^2 - 1)$ 個必要とするのに比べて、 $m$ 個の3入力1出力のセクタを必要とするだけで済み、必要な命令数、回路規模ともに大幅に削減できる。

【0054】図6は、本発明の第2の実施形態の構成を示した図である。図1に示した前記実施形態と比較すると、本実施形態においては、全プロセッサから同時にリードすることができ、ホストプロセッサ10またはプロセッサレイ1のうちの1つのプロセッサ2からライト

できるテーブルメモリ13が付加されている。

【0055】二値画像に対して、膨張処理を実行する際には、生成されたビットバックデータが「0」でないならば、周辺の8近傍画素の少なくとも1画素は「1」であるため、処理対象画素を「1」とすればよい。

【0056】これは、ビットバックデータに対する論理演算の最も簡単な演算の1つであり、このように簡単な論理演算で済む場合には、テーブルは必要ない。

【0057】これに対して、簡単な論理演算で処理できない場合には、ビットバックデータの値によって、予め計算した処理結果が格納されているテーブルデータを用意しておくテーブル参照法が有効である。テーブル参照法を実行するためには、テーブルとしてメモリには、プログラムのダウンロード時、またはテーブル参照法を実行する前までに、ビットバックデータの内容に対応するアドレスにそのビットバックデータに対する処理結果を書き込んでおく。

【0058】そして、テーブル参照処理では、ビットバックデータをアドレスとして、テーブルメモリを間接アドレッシングでアクセスすることにより、メモリから処理結果を読み出す。3×3画素領域のビットバックデータは8ビットであり、その取り得る値は「0」から「255」までの256通りであるため、この場合、テーブル参照用のメモリとして256ワードあれば良い。

【0059】複数PEにおいて、テーブル参照演算を実行するには、PE数分のテーブル参照メモリが必要とされることになるが、テーブルの内容は共通でよい場合が多く、各PE毎にテーブルを持たせる構成は、メモリの使用効率が悪い。

【0060】そこで、本実施形態においては、図6に示すように、全PEに対して、1つだけテーブルメモリ13を備え、全PEから同時にテーブルをアクセスするような構成とされている。

【0061】この場合、図7に示すように、テーブルメモリ13としては、1つのメモリセル30に対して、1つのライトポート31と、PE数( $c=n$ )分のリードポート32、すなわち $n$ 個のリードポート32を持つメモリを1つだけ用意すればよい。

【0062】ライトポート31は、ライトアドレス線と

ライトデータ線を介して、ホストプロセッサ10、コントローラ8、あるいは特定のPEと接続され、それらのいずれかからメモリセル30へのデータの書き込みを行う。

【0063】 $n$ 個のリードポート32の各ポートは、それ自身のリードアドレス線とリードデータ線とを介して、 $n$ 個のPEと個別に接続される。このように構成したことにより、全PEが、テーブルメモリのメモリセルを共有化することになり、高並列化した場合にも、回路面積(チップ面積)の増加を最小限に抑えることができ、一方、チップ面積に余裕がある場合には、より大容量のテーブルメモリを集積したり、あるいは複数面のテーブルメモリを集積することが可能とされ、その結果、テーブルメモリの1ワードのサイズを大きくしたり、複数のテーブルデータを同時に保持できることになり、テーブル参照処理の用途を拡大することができる。

【0064】回路的、あるいはチップレイアウト的な観点から、メモリのリードポートの数に限界がある場合には、全PEではなく、いくつかのPE毎に1つのテーブルメモリを割り当てればよい。例えば、全PE数が64で、8個のPE毎に1つのテーブルメモリを割り当てるならば、テーブルメモリとして、1つのライトポートと8つのリードポートを持つようにすればよいことになる。

【0065】

【発明の効果】以上説明したように、本発明によれば、プロセッサを多数内蔵する並列画像処理装置において、各プロセッサに $m$ 個の1ビットレジスタを設け、これらのレジスタは隣接するプロセッサから参照可能とし、各プロセッサが、 $(m-1)/2$  PE以内にあるプロセッサからの $m$ ビットずつのデータとそれ自身のレジスタの $(m-1)$ ビットの合わせて $(m^2-1)$ ビットを、格納されるレジスタに応じたビット位置に割り当て、 $(m^2-1)$ ビットレジスタにビットバックして格納できるようにし、さらに同時に $m$ 個の1ビットレジスタ間で、データをシフトし、シフトの際にデータが送られて来ない端部のプロセッサには、次のデータをメモリからロードできるように構成したことにより、二値画像データに対して、高速な画像処理を実行できる。

【0066】また、本発明によれば、ビットバックされたデータによるテーブル参照処理に用いられるメモリとして、1ライト/マルチリードポート構成のメモリを用い、全プロセッサからリードできるようにしたことにより、各プロセッサのローカルメモリをテーブルメモリとして用いる場合に比べて、大幅にメモリ使用量を削減することができる。

【図面の簡単な説明】

【図1】本発明の一実施形態の画像処理装置の構成を示すブロック図である。

【図2】二値画像の3×3近傍のビットバック処理を説



明するための図である。

【図3】本発明の一実施形態の画像処理装置におけるビットパックブロックの構成を示すブロック図である。

【図4】本発明の一実施形態の画像処理装置の両端部のプロセッサにおけるビットパックブロックの構成を示すブロック図である。

【図5】本発明の一実施形態の画像処理装置に使用されるビットパック命令を説明するための図である。

【図6】本発明の他の実施形態の画像処理装置の構成を示すブロック図である。

【図7】本発明の他の実施形態の画像処理装置におけるテーブルメモリの構成を示すブロック図である。

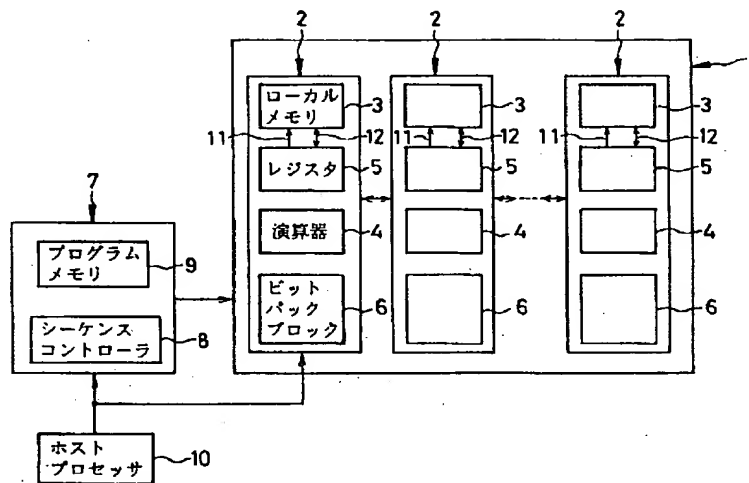
【図8】従来の画像処理装置の構成を示すブロック図である。

#### 【符号の説明】

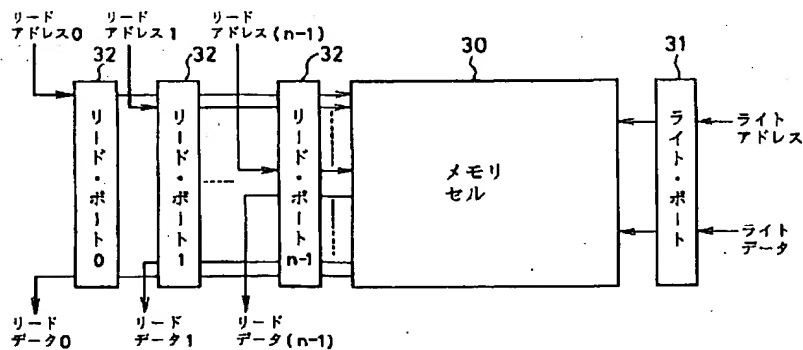
- 1 プロセッサアレイ
- 2 プロセッサ
- 3 ローカルメモリ
- 4 演算器

- \* 5 レジスタ
- 6 ビットパックブロック
- 7 命令供給回路
- 8 シーケンスコントローラ
- 9 プログラムメモリ
- 10 ホストプロセッサ
- 11 アドレス線
- 12 データ線
- 13 テーブルメモリ
- 10 20 データ転送線
- 21 1ビットレジスタ
- 22 セレクタ
- 23 デスティネーション・レジスタ
- 24 セレクタ
- 30 メモリセル
- 31 ライトポート
- 32 リードポート
- 100 一次元プロセッサアレイ
- \* 200 プロセッサ

【図1】



【図7】



【図2】

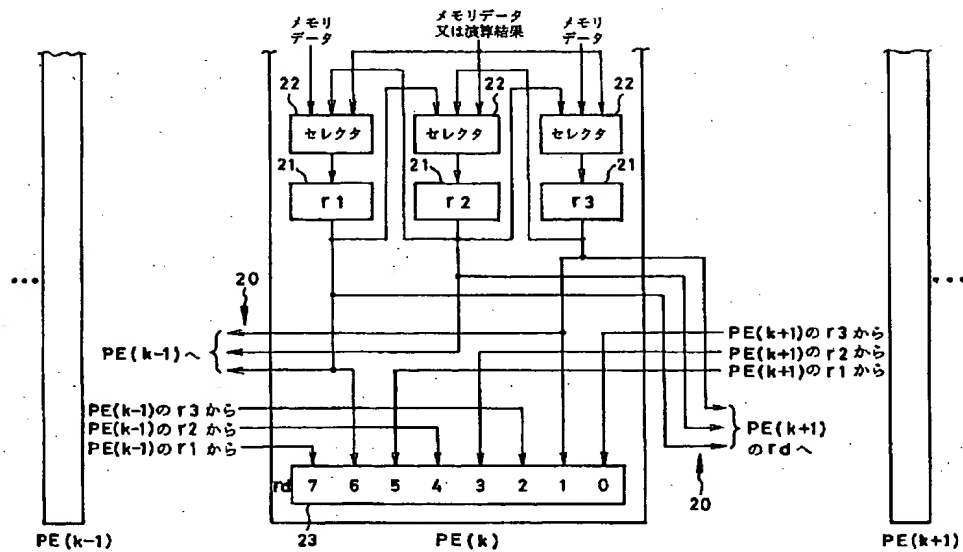
(A)

	(k-1)列	k列	(k+1)列
(m-1)行	$P(k-1, m-1)$	$P(k, m-1)$	$P(k+1, m-1)$
m行	$P(k-1, m)$	$P(k, m)$	$P(k+1, m)$
(m+1)行	$P(k-1, m+1)$	$P(k, m+1)$	$P(k+1, m+1)$

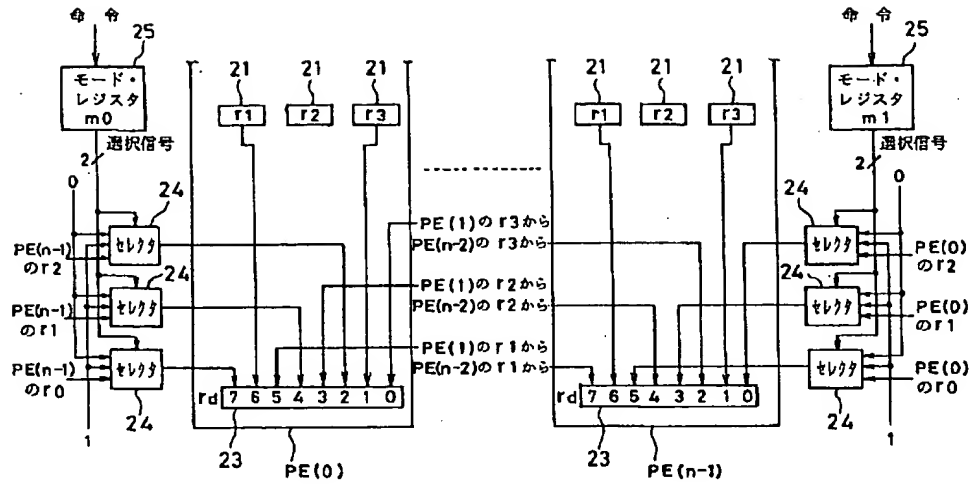
(B)

7	6	5	4	3	2	1	0ビット目
$P(k-1, m-1)$	$P(k, m-1)$	$P(k+1, m-1)$	$P(k-1, m)$	$P(k, m)$	$P(k+1, m)$	$P(k-1, m+1)$	$P(k+1, m+1)$

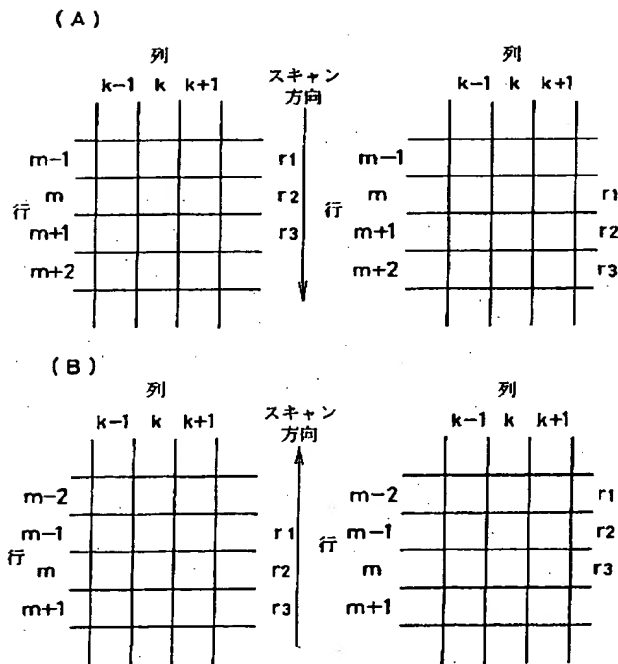
【図3】



【図 4】



【図 5】



[illegible]